Towards a Discipline for Evaluating Ubiquitous Computing Applications

J. Scholtz, S. Consolvo

IRS-TR-04-004

January 2004

Intel Research

Towards a Discipline for Evaluating Ubiquitous Computing Applications

Jean Scholtz[1], Sunny Consolvo[2]

[1] National Institute of Standards and Technology
jean.scholtz@nist.gov
[2] Intel Research
sunny@intel-research.net

Abstract. Though evaluations are being performed on ubicomp applications, it is difficult for researchers to learn from each other's results. We believe that this is because no framework exists for conducting user evaluations of ubicomp applications. In this paper, we propose a framework in hopes of making it easier for researchers to learn from each other's results, to create effective discount evaluation techniques and design guidelines for ubicomp, to provide a mechanism for researchers to share what they have learned about the appropriateness of different evaluation techniques, and to provide structure so that key areas of evaluation are not overlooked.

1 Introduction

User evaluations are conducted to assess the quality of an application or to help the research team design, refine, or determine requirements for an application. For useful and usable ubiquitous computing (ubicomp) applications to be produced within a reasonable timeframe, it is important that researchers be able to learn from results of each other's evaluations. Though evaluations are currently being performed on ubicomp applications and subsequently published, it is difficult for the researchers to learn from each others' results. This may be because published results focus on evaluations conducted to assess the quality of a particular application, but it may also be because there is not currently a shared terminology being used throughout the field. For example, if a researcher is interested in learning about issues of *trust* in ubicomp, it is difficult to quickly find the relevant results from a variety of publications.

We believe that the way to improve results sharing in the field is to create a user evaluation framework specifically for ubicomp. Frameworks create structure which ensures that key areas of importance are not overlooked in evaluations. They also establish sets of terms that are used to describe results. By using the same terminology when publishing results, researchers should be able to learn from each others' results. Results sharing should lead to the establishment of design guidelines and sets of evaluation techniques that can be used to investigate different evaluation areas. It

should also lead to the development of ubicomp-specific discount evaluation techniques to enable quicker and less costly evaluations.

Our contribution in this paper is to lay the groundwork for establishing a framework for evaluating ubicomp applications. Our primary goals are: (1) make it easier for researchers to learn from each other's evaluations, (2) enable the creation of effective discount evaluation techniques and design guidelines specifically for ubicomp, (3) provide a mechanism for researchers to share the appropriateness of different evaluation techniques for exploring specific areas of evaluation, and (4) provide structure to evaluators so that key areas are not overlooked in their evaluations.

We begin with a discussion of related work. We follow with a discussion of user evaluations for desktop computing in an attempt to show why some, but not all, of desktop computing's design guidelines, metrics, and evaluation techniques can be used for ubicomp. We then discuss the model for ubiquitous computing and propose our ubicomp user evaluation framework. We close with a discussion of future work and our conclusion.

2 Related Work

Attempts have been made to start creating structure in ubicomp, but none are complete. Some focus on subsets of ubicomp, such as sensing systems. Others focus solely on areas like values. Our proposed framework encompasses the field of ubicomp and is meant as a tool for evaluators. It follows the same spirit as the following work, but is trying to create a structure for the entire field of ubiquitous computing. All of the discussed works address important design and evaluation issues for different areas of computing research. Where appropriate, their suggestions have been incorporated into our framework.

Jameson [Jameson03] proposes five usability challenges for adaptive interfaces: (1) predictability and transparency, (2) controllability, (3) unobtrusiveness, (4) privacy, and (5) breadth of experience. Jameson's work focuses solely on *adaptive interfaces* (i.e., systems that learn from the user's behavior and react accordingly) and *usability*[1] (e.g., though *privacy* is represented in his challenges, *trust* is not). Our framework encompasses the field of ubiquitous computing and addresses evaluation areas including, but not limited to, usability.

Bellotti et. al. [Bellotti02] suggest five interaction challenges for designers and researchers of sensing systems: (1) address—"directing communication to a system," (2) attention—"establishing that the system is attending," (3) action—"defining what is to be done with the system," (4) alignment—"monitoring system response," and (5) accident—"avoiding or recovering from errors or misunderstandings." Bellotti focuses on challenges for the system designer and on communicative aspects of interaction in sensing systems (specifically, interactions that are non-Graphical User Interface (GUI) based). Our framework is targeted at the evaluator, does not assume a particular style of interaction, and is not limited to interactions. It also encompasses

[1] Friedman and Kahn discuss the distinction between *usability* and *values* [Friedman03, pp.1180-1].

the field of ubicomp in general, not just sensing systems (e.g., text messaging is arguably ubicomp, but does not involve sensing).

Friedman and Kahn [Friedman02] suggest 12 key human values with ethical import: (1) human welfare, (2) ownership and property, (3) freedom from bias, (4) privacy, (5) universal usability, (6) trust, (7) autonomy, (8) informed consent, (9) accountability, (10) identity, (11) calmness, and (12) environmental sustainability. Friedman's values are for the entire field of Human-Computer Interaction (i.e., including websites) and focus on design considerations. Usability issues, such as *interaction*, are not represented.

Though much about evaluating ubicomp can be learned from desktop computing research, there are key differences that necessitate a framework specifically for ubiquitous computing. We now discuss user evaluations for desktop computing.

3 User Evaluations for Desktop Computing

Traditional desktop computing applications are based on the model of one user per application at any given time. The typical environment is at an office or home with the user seated at a desk. He is using one monitor, with a keyboard and mouse as interaction modalities. In special cases, he may use speech or a pen for input.

Competition for the user's attention is assumed to be low and is usually not duplicated in usability testing. This competition can come from interruptions caused by a telephone call or a co-worker/family member stopping by. Other applications on the desktop may also compete for the user's attention. An incoming email may cause an alert to sound. A notification of a meeting or the arrival of an instant message may interrupt the user. The majority of these interruptions can be controlled by the user. Many applications that deliver notifications allow the user to specify if and how they wish to be notified of an event. Users can close their door to control interruptions by co-workers and family members. Some users attach "rear view mirrors"[2] to their monitors so they are not startled when approached from behind**.** In noisy office environments, users may listen to music over headphones. Testing in usability laboratories does not usually include these types of disruptions. However, field studies aim to uncover the ease with which users can recover from such interruptions.

Currently, usability evaluations focus on three metrics: efficiency, effectiveness, and user satisfaction (ISO 92411-11). *Efficiency* measures the amount of time users take to perform a particular task. *Effectiveness* measures the percentage of that task the majority of users are able to complete with and without assistance. *Satisfaction* measures are obtained from users' ratings of their interactions with the application.

---

[2] Some examples of monitor mirrors are *Air Technologies Corporation's Computer Monitor Mirror*, (http://shop.store.yahoo.com/airtechcorp/21monmir.html), and *Feng Shui Warehouse's Monitor Mirror* (http://fengshuiwarehouse.net/mirrors.html)

There are two distinct types of user evaluations: formative and summative. *Formative* evaluations [Nielsen93, p.170] are conducted "to help improve the interface as part of an iterative design process." *Summative* evaluations "aim at assessing the overall quality of an interface." While summative evaluations are always empirical, formative evaluations include techniques such as usability inspection methods (e.g., heuristic evaluation and cognitive walkthrough), formal modeling techniques (e.g., GOMS—Goals, Operators, Methods, and Selection Rules), and paper prototyping studies [Nielsen94]. A number of user evaluation techniques have been developed for desktop computing applications. Figure 1 shows a rough timeline of highlights in the development of desktop computing evaluations from 1971 - 2001.

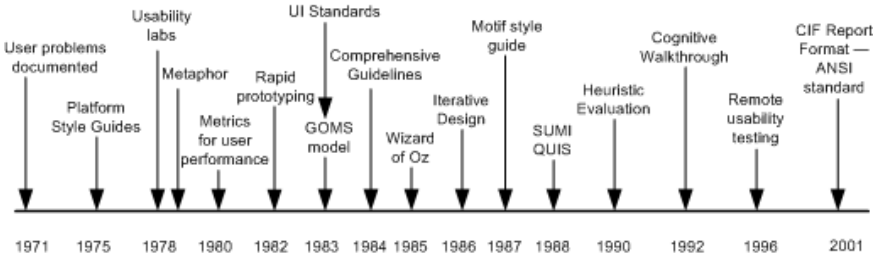

Fig. 1. 30 years of highlights in the development of desktop computing user evaluations from 1971 - 2001

The development of guidelines for desktop computing systems has had an enormous effect on the implementation of more usable systems. Usability inspection methods have been based on such guidelines[3]. Interaction widgets have been developed and are now available as toolkits in most platform development systems. This allows designers to achieve consistency in look and feel as well as behavior across different applications. By using guidelines when making design choices, a reasonable first attempt at a user interface can be produced, thereby alleviating the need for extensive evaluation. That is, designers of desktop applications do not have to "start from scratch" every time they design a new application. With the advent of web applications and web sites, remote user testing is becoming more popular. This allows empirical testing to be performed with more users over a larger geographical area at a lower cost. Though usability and user interface toolkits have come a long way in desktop computing and continue to be refined, the field is starting to place more of an emphasis on values, emotion, privacy, trust, and the social aspects of computing[4].

Currently, designers of ubiquitous computing systems must often "start from scratch" when it comes to designing the user experience. It is our hope that because

[3] Usability inspection methods, such as the Heuristic Evaluation and Cognitive Walkthrough, emerged almost 20 years after the first user problems were documented

[4] At CHI 2003, sessions included the following topics: *Trust, Security, & Safety, Privacy & Trust, Digital Sociability, Design for the Socially Mobile,* and *Emotions.*

we have the benefit of learning from desktop computing (specifically, what was required to develop effective discount usability techniques and design guidelines), that we will be able to move quickly forward in the area of user evaluations for ubicomp. We are also inspired by the value sensitive design work of Friedman et. al. and want to incorporate areas beyond usability, such as values, into our proposed framework.

4 User Evaluations for Ubiquitous Computing

One of the first things researchers may ask is why the design guidelines, metrics, and evaluation methodologies from desktop computing cannot be used "as is" for ubicomp. While a number of evaluation methods, metrics, and design guidelines can be borrowed for ubicomp, there are considerable differences in the models of these two types of computing that suggest different evaluation methodologies as well as metrics.

First, we need to consider the model for ubicomp applications.

4.1 Ubiquitous Computing Model

Before we talk about a framework for evaluating ubicomp applications, we must first define what we mean by ubicomp. Weiser's [Weiser91] vision of ubicomp was of computing so integrated into everyday objects that it becomes invisible to users. Today, ubicomp applications are diverse in nature, ranging from small applications that help commuters track train and bus schedules [Lunde01] to smart laboratories [Arnstein02], smart museums [Fleck02], and instrumented classrooms [Abowd99]. Moran and Dourish [Moran01] note that what is common to the various ubicomp efforts is that "they move the site and style of interaction beyond the desktop and into the larger real world where we live and act." They go on to suggest that "the design challenge, then, is to make computation useful in the various situations that can be encountered in the real world—the ever changing context of use." Along this line is the concept that the application is secondary to other tasks the user is performing. Though this goal is shared by desktop computing, the differences between the computing environments mean different and often more serious implications for ubicomp. This design challenge and the implications for ubicomp motivate our user evaluation framework.

The ubiquitous computing environment may contain many devices with which the user interacts. Speech, gestures, and even physical interactions with devices can be used as interaction modalities. In some cases, the user may not need to consciously do anything. Likewise, the feedback to users is not limited to one particular display, or in fact to any display. Behavior by the user may cause actions in the physical world. For example, lying down in an intelligent room can cause the drapes to close, the lights to dim, and the music to be turned off [Brooks97]. Both input and output in a ubicomp environment may be distributed.

Additionally, as ubicomp occurs everywhere, there may be a number of users interacting with a system simultaneously [Fleck02]. This necessitates the question of how the interactions of one user might affect another user and if/how ubiquitous computing impacts the normal social situation. As with desktop computing, there is the need to consider both direct and indirect stakeholders [Friedman01]. "*Direct stakeholders* refer to parties – individuals or organizations – who interact directly with [the system] or its output. *Indirect stakeholders* refer to all other parties who are affected by the use of the system. Often, indirect stakeholders are ignored in the design process." For ubicomp applications to become adopted by the general public, it is crucial for evaluators to consider *all* stakeholders, not just direct.

A number of ubiquitous computing applications are "context-aware." That is, the behavior of the application changes based on what the user is doing. Dey [Dey2001] defines context as "any information that characterizes a situation related to the interaction between humans, applications, and the surrounding environment." In practice, different types of sensory input are used to infer context. User location is a popular contextual attribute used in a number of context-aware applications such as mobile tour guides [Abowd97, Feiner97].

4.2 The Current State of User Evaluations in Ubicomp

Evaluation of ubiquitous computing applications is currently a labor-intensive chore. First, evaluations are carried out on a prototype of the application. This means that a robust prototype has to be developed and deployed, and though it doesn't have to be product-quality, it has to be reasonably safe (e.g., no sharp edges). Considerable development work has to be done to accomplish this, decreasing the willingness of the research team to make significant changes uncovered by evaluations. In some cases, Wizard-of-Oz techniques may be used, though the "reasonably safe" requirement still applies. Ubicomp applications currently involve customized infrastructure, environments, and/or devices. This often means it is difficult to conduct evaluations with large numbers of users (e.g., it may be too time consuming or cost prohibitive to produce more than a few prototypes of a device) and/or with several groups of users (e.g., though a study may be conducted with several inhabitants of an office in an instrumented space, it may be difficult to duplicate the study at other offices). Reasons such as these emphasize the importance of performing formative evaluations before any (or at least before significant) development occurs. Secondly, evaluations of ubiquitous computing applications are extremely diverse. Researchers conduct evaluations specific to their application and report results using their own terms to describe what they evaluated, making it difficult for other researchers in the community to use the lessons learned, or even be able to apply the same evaluation techniques.

Our premise is that identification of a set of areas for evaluation, along with suggested metrics and measures for ubiquitous computing applications would advance the field. Though researchers would select the measures appropriate for their particular application, having a standard framework from which to work and a standard set of terms to use should enable researchers in the field to learn from each other's re-

sults. It should also enable others who are interested in evaluating the same metrics on their own applications learn about the evaluation techniques they might use to conduct their studies. As we build up knowledge of the properties needed to ensure the success of ubiquitous computing systems, we will be able to develop design guidelines and lower-cost evaluation methodologies, as in the world of desktop computing. A recent work by Mankoff et. al. [Mankoff03] has identified heuristics for ambient displays. While this work touches only a small portion of ubiquitous computing, we are encouraged and confident that many other aspects of ubiquitous computing can benefit from similar work. We also hope that having a framework from which to work will ensure that key areas of evaluation are not overlooked.

4.3 A Proposed Framework for User Evaluations of Ubicomp

We have developed a set of areas for evaluation, along with sample metrics and measures. We call these "Ubicomp Evaluation Areas" (UEAs). They have been assembled from personal experience in evaluation efforts and a literature review. In our framework we present metrics and conceptual measures. A measure is an observable value. A metric associates meaning to that value by applying human judgment. Metrics can be composite; that is, they are interpretations of one or more contributing elements, e.g., measures or other metrics. We use the term conceptual measure here as opposed to implementation specific measure. An evaluator using this framework will have to decide how a particular conceptual measure can be collected. That instantiation becomes the implementation specific measure. Measures can be both quantitative and qualitative. They can also be directly observable or indirectly measured. As the framework is used, we will obtain different implementation-specific measures that can be shared with others in the community.

To conduct evaluations using the following framework, evaluators must begin by identifying users who will be affected by the application. Friedman et. al. [Friedman01] define these as direct and indirect stakeholders. *Direct stakeholders* interact with the application and/or its output in a direct way. *Indirect stakeholders* are affected by the application in a meaningful, but not direct way. For example, the direct stakeholder (DS) of a cell phone is the person who uses the cell phone and makes/receives calls from it. The indirect stakeholders (IS) of the cell phone include:

- people who receive calls from the DS
- people who call the DS
- people who are with the DS when he uses his cell phone
- people around (but not "with") the DS when he uses his cell phone (including people driving near the DS if he happens to be using his phone while driving)

It is important to understand that one person may be both a direct and an indirect stakeholder for the same ubicomp application. For example, a person may be a direct stakeholder when he is using his cell phone and an indirect stakeholder when he is with someone who is using her cell phone. In fact, he may be both at once if he receives a call on his cell phone that was made by someone from her cell phone. Once

the direct and indirect stakeholders have been identified, the researcher may decide which stakeholders should be involved in different aspects of the evaluation(s).

The evaluator must also decide if she needs to establish a baseline or control group. This will give the evaluator a means of comparing the technology she is evaluating to the user's normal environment.

For each UEA we offer a definition, brief discussion, sample metrics and measures, and an example(s) from desktop or ubiquitous computing as appropriate. We expect that this framework will be refined as it is used by the community.

UEA 1: Attention

Attention is defined [Proctor03] as "increased awareness directed at a particular event or action to select it for increased processing." The idea of *attention* has been explored in depth in the area of desktop computing. Nielsen [Nielsen93, pp.135-7] reminds us of important time limits to consider when providing the user with feedback. Bly and Rosenberg [Bly86] investigated tiled versus overlapping windows to determine which arrangement was more efficient for users. Early studies also looked at different ways to "grab" the user's attention and derived guidelines for the use of highlighting and color [Smith86, Galitz85]. Norman [Norman88, pp.164-5] breaks from the desktop computing model and uses toasters to describe "selective attention" (i.e., a type of tunnel vision), an important consideration for designers. Thanks to studies such as these, desktop computing designers know how to deal with many attention issues. For example, current windowing system toolkits include appropriate ways to handle overlapping windows.

However, attention is likely to be more of an issue for ubicomp, as users are handling other physical or mental tasks in parallel to interacting with ubicomp devices. They may be using those devices in a variety of environments, with a variety of different people nearby.

Metrics for *Attention* include focus and overhead. *Focus* refers to where the user is directing their attention. Focus is extremely important in ubiquitous computing as numerous devices may be involved. The user may have to shift focus between devices a number of times to accomplish an interaction or to determine that progress is being made. As part of the evaluation of Labscape, Consolvo et. al. [Consolvo02] used Lag Sequential Analysis to look at focus. Their premise was that the more interleaved Labscape and "regular work" were, the more likely it was that Labscape was being smoothly integrated into the environment, and therefore, the more the biologists were able to focus on the biology and not the new technology.

Overhead refers to the any "wasted time" introduced by the technology. For example, evaluators could measure the amount of time the user spends switching between the technology and other foci. She could compare the time it takes the user to complete the task with and without the technology. The evaluator could also ask the user for his opinion of what it was like to perform the task with and without the technology. Reiners et. al. [Reiners99, p.32] used augmented reality techniques to help with the task of assembling a door lock into a car door. They claim that "three-dimensional animated instructions can be integrated into the surrounding environment at the exact place where the action has to be performed so that no mental transfer is needed." Curtis et. al. [Curtis, p.48] describe how they tried to increase the produc-

tivity of wire bundle assembly at Boeing by imbedding the relevant information into the physical display where it was needed. Traditionally, the worker would have to refer to an instruction sheet that accompanied the set of wires he needed to assemble.

Metric: Focus
Conceptual measures:

- Number of times a user needs to change focus due to technology
- Number of different displays/ actions a user needs to reference to accomplish an interaction or to check on the progress of an interaction
- Number of events not noticed by a user in an acceptable time
- Workload imposed on the user attributable to focus

UEA 2: Adoption

Few evaluators have looked at adoption as an area for evaluation. Grudin [Grudin88] discussed adoption in reference to CSCW applications and notes that a critical mass is needed for collaboration technologies to be useful and successful. Moore discusses how technology is adopted and points out the value of having a referent to observe to determine the utility of the technology before adopting it [Moore91]. Downes and Mui [Downes98] give 12 rules for designing radical technologies. Two of these are applicable to measures for adoption: user continuity and user sacrifice. Electronic shopping is an example of user continuity and vendor disruption. To a customer used to catalog shopping, electronic shopping is a reasonable extension of this. The user can get essentially the same service but in a shorter time frame using the web as she could by mailing in an order. On the vendor's side of the house, there are considerable differences needed for implementation. User sacrifice refers to the services or value that the user actually gets compared to what the user really wanted.

The metrics and measures for adoption are of two types – those that measure the actual adoption and those that help to predict the success or failure of the application. Categories and sample metrics for *Adoption* are rate, value, and availability.

Metric: Rate
Conceptual Measures:

- New users/unit of time
- Adoption rationale
- Technology usage statistics

Metric: Value

NOTE: When investigating *value*, it is important to consider all stakeholders.

Conceptual Measures:

- Change(s) in productivity
- Perceived cost/benefit
- Continuity for user
- Amount of customer sacrifice

Metric: Availability
Conceptual Measures:

- Number of actual users from each target user group
- Technology supply source
- Categories of users in post-deployment

UEA 3: Trust

Awareness of other users and their activities is important in multi-user systems to facilitate coordination of tasks and resources. Dourish and Bellotti [1992]defined awareness as "an understanding of the activities of others, which provides a context for your own activities." On the other hand, privacy is also an issue in multi-user systems. The more information is shared, the more awareness can be increased but at a cost in privacy. For ubicomp applications trust is directly related to awareness and privacy. When a user interacts with ubicomp applications such as tour guides, there is definite value in knowing what venues other users found interesting. On the other hand, having information saved about your visits may be disconcerting to users concerned with privacy issues. Drury [2001] has developed the Synchronous Collaborative Awareness and Privacy Evaluation (SCAPE) heuristic evaluation methodology for collaborative applications. SCAPE provides both a means of specifying awareness and privacy requirements and evaluating whether the application satisfies these requirements. It may be feasible to adapt evaluation methodologies such as SCAPE to ubicomp applications.

Metric: Privacy
Conceptual Measures:

- Amount of information user has to divulge to obtain value from application
- Availability of explanations to user about use of recorded data

Metric: Awareness
Conceptual Measures:

- Ease of coordination with others in multi-user application
- Number of collisions with activities of others

UEA 4: Conceptual Models

A conceptual model [Mullet02] provides the basis for understanding an interactive device or program. It names and describes the various components and explains what they do and how they work together to accomplish tasks. Understanding the conceptual model makes it possible to anticipate the behavior of the application, to infer "correct" ways of doing things, and to diagnose problems when something goes wrong.

Different kinds of models exist to meet different needs. Though designers and developers have different conceptual models for the same application, for the purposes of this paper, we are interested in the *user's conceptual model*. For example, analogies or metaphors, such as the desktop metaphor, offer affordances in support of

conceptual models. The distributed nature of ubiquitous computing makes it challenging for users to build unified models of behaviors and interactions. For example, how does a user know when they are in a "smart room?" When the user is in a smart room, will they know how to interact with the room?

Scholtz and Bahrami [Scholtz 03] conducted an experiment to assess how well users could create mental models of robot behavior. This experiment was part of research efforts to develop evaluation techniques for various roles of interaction in human-robot interaction (HRI) [Scholtz01]. The users were playing the bystander role – no instruction in interaction was given to them. The users were first asked how they thought they could interact with the robot. Users were then assigned to one of four conditions. In these conditions the robot behavior was either expected or unexpected, and consistent or inconsistent. As the robot in this case was a dog-like robot, dog-like behaviors were expected. Inconsistency was generated by randomizing actions in response to interactions. After the users were told what interactions were possible, they were given some time to play with the robot. They were then asked what interactions produced which actions. Not surprisingly, users were more successful at forming mental models when the actions of the robot were expected and consistent.

Metric: Predictability of application behavior
Conceptual Measures:

- Degree of match between user's model and actual behavior of the application

Metric: Awareness of application capability(ies)
Conceptual Measures:

- Degree of match between user's model and actual functionality of the application

Metric: Vocabulary awareness
Conceptual Measures:

- Degree of match between user's model and the syntax of multimodal interactions.

UEA 5: Interaction

As previously discussed, usability evaluations in the desktop world use the metrics of effectiveness, efficiency, and user satisfaction. While these three metrics are also applicable to interactions in ubiquitous computing, evaluations must take into consideration differences between desktop and ubiquitous computing. Shafer [Shafer01] suggest these differences:

- interactions in ubiquitous computing can be physically embedded
- the set of input and output devices are dynamic rather than static as in desktop systems
- as multiple devices are used, there is no single focal point
- there can be multiple simultaneous users

Additional measures are needed to evaluate these aspects of ubiquitous computing. Guidelines have been developed for the design of graphical user interactions based on mouse and keyboard input and a single display as output. The ubicomp community needs studies and evaluations for distributed, multimodal interactions in a ubiquitous computing environment.

Metric: Effectiveness
Conceptual Measures:

- percentage of task completion

Metric: Efficiency
Conceptual Measures:

- time to complete a task

Metric: User Satisfaction
Conceptual Measures:

- user rating of performing the task

Metric: Distraction
Conceptual Measures:

- Time taken from the primary task
- Degradation of performance in primary task
- Level of user frustration

Metric: Interaction transparency
Conceptual Measures:

- Effectiveness comparisons on different sets of input/output devices.

Metric: Collaborative interaction
Conceptual Measures:

- Number of conflicts
- Percentage of conflicts resolved by the application
- User feelings about conflicts and how they are resolved
- User ability to recover from conflicts

Collaborative interaction is focused on measuring only the interactions. The Trust UEA also looks at collaboration aspects but from the aspect of what information is available to users in multi-user systems of activities of other users.

UEA 6: Invisibility
"Smart" ubicomp applications (i.e., context-aware applications) make inferences about the user's activities, goals, emotional state, and social situation and attempt to act on behalf of the user. If the system has sensed and interpreted the context correctly, this initiative can result in time savings and a reduction in user workload. However, if the system has misjudged the situation, the user may have to intervene.

This may result in a cost in time, in embarrassment to the user, and even a potentially dangerous situation. Bellotti [Bellotti01] maintains that context-aware systems need to be intelligible and accountable. Systems that sense and use context need to explain that understanding of context to users who can then judge the accuracy. Users are ultimately responsible for the actions of the system, therefore this notion of accountability must be designed into the system.

Smart systems may also allow users to customize how the system responds based on their personal preferences. Users may be asked to explicitly input this information or the system may learn preferences based on a series of interactions.

Evaluation of context-aware systems can be extremely time consuming if the system uses many context variables. How can all of these combinations of state be tested with users? An interesting methodology is being used by [Bylund02] which relies on using 3D simulations of the environment to show users how the system reacts to various contextual settings.

Metric: Intelligibility
Conceptual Measures:

- User's understanding of the system explanation

Metric: Control
Conceptual Measures:

- Effectiveness of interactions provided for user control of system initiative.

Metric: Accuracy
Conceptual Measures:

- Match between the system's contextual model and the actual situation.

Metric: Appropriateness of action
Conceptual Measures:

- Match between the system action and the action the user would have requested.

Metric: Customization
Conceptual Measures:

- Time to explicitly enter personalization information or time for the system to learn and adapt to the user's preferences.

UEA 7: Impact

The last ubicomp evaluation area is that of impact. Even well-designed technology does not always succeed. At times that is not a function of the actual application but of unintended consequences or side effects of the system. During an evaluation conducted by one of the authors the root cause of the non acceptance of a system was determined. While the system as implemented certainly needed usability improvement, a more serious issue was the role change that that was imposed by the system.

The users of the system were being called upon for information that they did not have.

Social acceptance plays a role in whether technology is used. Curtis noted that users of the Boeing wiring system were not comfortable being seen by others in the company while they were wearing "socially unacceptable" goggles needed for using the system [Curtis99]. How many people today really want to walk down the street wearing a computer on their belt with small displays over one eye?

Metric: Behavior changes
Conceptual Measures:

- Type, frequency and duration
- Match between user's current job description and application role

Metric: Social acceptance
Conceptual Measures:

- Requirements placed on user outside of social norms

Metric: Environment change
Conceptual Measures:

- Type, frequency and duration

4.4 Interpretation of UEA Metrics

In typical desktop usability evaluations the measures of effectiveness, efficiency, and user satisfaction are not considered equally. If the software application being evaluated is used in a discretionary fashion, more emphasis may be placed on user satisfaction. If the application is used in time critical situations or in situations where seconds saved may add up to thousands of dollars saved, the efficiency of the interactions should be optimized. When errors are not tolerated, as in life critical situations, then the effectiveness metric is the most important measure.

The same idea applies to ubicomp, however more measures may be applicable. Evaluators and other team members must decide which measures are the most critical for the type of application being evaluated. These decisions must consider the environments in which the application will be used and the needs of all stakeholders.

How should evaluators prioritize the various UEAs for ubiquitous computing? While it is too early to say definitely, we can make some predictions:

- Any applications that are designed to be "walk-up and use" will have to score well in metrics related to interaction and conceptual models.
- Applications that are developed to be used in a social setting, in addition to scoring well for interaction metrics and conceptual models, will need good scores in impact.
- Applications that deal with personal information of users will certainly need high scores in trust.
- If the ubicomp application is targeting users involved in a time or life critical situation, interaction and attention will be of utmost importance.

- Context-aware applications may score low on measures of predictability and conceptual models, but high on efficiency and effectiveness.

5 Future Work

While this framework is based on evaluations from desktop computing, a ubicomp literature review and personal experience, we are in the process of applying it to an evaluation of a ubicomp application and have plans to apply it to future evaluations of other applications. We expect that the framework will be refined as we and other researchers use it. We caution that the seven areas of the framework are by no means independent. We are also interested in looking at the interactions between the different areas. For example, how does a low score in the conceptual model area affect the ubicomp area of trust? These are interesting questions as one ubicomp area may be easier to evaluate than another. Therefore finding indirect measures may make evaluation more feasible.

6 Conclusion

We have presented a framework for conducting user evaluations on ubiquitous computing applications. The framework builds on techniques, metrics, guidelines, and new trends from desktop computing as well as some more recent "challenges" proposed for specific areas of ubicomp. The goals of the framework are: (1) make it easier for researchers to learn from each other's evaluations, (2) help create effective discount evaluation techniques and design guidelines, (3) provide a mechanism for researchers to share the appropriateness of different evaluation techniques for exploring specific areas of evaluation, and (4) provide a structure to ensure that key areas of evaluation and indirect stakeholders are not overlooked.

7 Acknowledgments

We would like to thank the following colleagues for their support and inspiration: Bill Schilit, Larry Arnstein, Tim Kindberg, Miryung Kim, Kevin Mullet, James Landay, Gaetano Borriello, Batya Friedman, Anthony LaMarca, and Steve Gribble.

References

[Abowd97] Abowd, G. D., Atkeson, C. G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. 1997. "Cyberguide: A mobile context-aware tour guide." *ACM Wireless Networks*, 5, 421-433.

[Abowd99] Abowd, G. D., "Classroom 2000: An experiment with the instrumentation of a living educational environment," *IBM Systems Journal*, Vol, 38. (1999)

[Arnstein02] Arnstein, L.F., Borriello, G., Consolvo, S., Franza, B.R., Hung, C., Su, J., Zhou, Q,H. 2002. "Labscape: Design of a Smart Environment for the Cell Biology laboratory," *IEEE Pervasive Computing*, vol 1(3), (2002), pp.13-21.

[Bellotti01] Bellotti, V. and Edwards, K. 2001. "Intelligibility and Accountibility: Human Considerations in Context-Aware Systems," *Human-Computer Interaction*, vol. 16, 2-4, (2001), pp.193-212.

[Bellotti02] Bellotti, V., Back, M., Edwards, W.K., Grinter, R.E., Henderson, A., Lopes, C., "Making Sense of Sensing Systems: Five Questions for Designers and Researchers," *Proceedings of the Conference on Human Factors and Computing Systems* (2002) pp. 415-422.

[Bly86] Bly, S. A. and Rosenberg, J. K. "A comparison of tiles and overlapping windows," *Proceedings of the CHI/86 Human Factors in Computing Systems Conference*. New York: Association for Computing Machinery, (1986), pp.101-6.

[Brooks97] Brooks, R. A., "The Intelligent Room Project," *Proceedings of the Second International Cognitive Technology Conference (CT'97)*, Aizu, Japan, (August 1997).

[Bylund02] Bylund, M. and Espinoza, F., "Testing and demonstrating context-aware services with Quake III Arena", *Communications of the ACM*, ACM Press, New York, (2002), pp. 46-8.

[Consolvo02] Consolvo, S., Arnstein, L., and Franza, B., "User Study Techniques in the Design and Evaluation of a Ubicomp Environment," *Proceedings of the Fourth International Conference on Ubiquitous Computing*, (September 2002), pp. 73-90.

[Curtis99] Curtis, D., Mizell, D., Gruenbaum, P., Janin, A., "Several Devils in the Details: Making an AR Application Work in the Airplane Factory," *Augmented Reality: Placing Artificial Objects in Real Scenes, Proceedings of IWAR'98*, AK Peters, Massachusetts (1999) p.48.

[Dey01] Dey, A., Abowd, Gregory, and Salber, D., "A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications," *Human-Computer Interaction*, vol. 16, 2-4. (2001), pp.97-166.

[Dourish92] Dourish, P. and V. Bellotti. Awareness and Coordination in Shared Workspaces. *CSCW '92*, Toronto, Canada, ACM.(1992).

[Downes98] Downes, L. and Mui, C. Boston, MA: Harward Business School Press. (1998).

[Drury01] Drury, J. Extending usability inspection evaluation techniques for synchronous collaborative computing applications. Sc.D. thesis, Department of Computer Science, University of Massachusetts Lowell. (2001).

[Feiner97] Feiner, S., MacIntyre, B., Hollerer, T., Webster, A. "A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment," *Personal Technologies*, 1, (1997), pp. 208-217.

[Fleck02] Fleck, M., Frid, M., Kindberg, T., O'Brien-Strain, E., Rajani, R., and Spasojevic, M. 2002. "Rememberer: A Tool for Capturing Museum Visits," *Proceedings of the Fourth International Conference on Ubiquitous Computing*, (September 2002), pp. 48-55.

[Friedman01] Friedman, B., Kahn, Jr., P.H., Borning, A., "Value Sensitive Design: Theory and Methods," University of Washington Technical Report 02-12-01 (December 2001).

[Friedman03] Friedman, B., Kahn, Jr., P.H., "Human Values, Ethics, and Design," *The Human-Computer Interaction Handbook*, Lawrence Erlbaum Associates, New Jersey (2003) pp. 1177-1201.

[Galitz85] Galitz, W. O. *Handbook of screen format design*. Wellesley Hills, MA: QED Information Sciences, (1985)

[Grudin88] Grudin, J. (1988). Why CSCW Applications Fail. *CSCW 88*, ACM.

[Hudson96] Hudson, S. and I. Smith (1996). Techniques for Addressing Fundamental Privacy and Disruption Tradeoffs in Awareness Support Systems. *CSCW 96 Conference on Computer Supported Cooperative Work*, Cambridge, MA, ACM.

[Jameson03] Jameson, A. "Adaptive Interfaces and Agents," *The Human-Computer Interaction Handbook*, Lawrence Erlbaum Associates, New Jersey (2003) pp. 316-318.

[Lunde01] Lunde, T. and Larsen, A. "KISS the Tram: Exploring the PDA as Support for Everyday Activities," *Proceedings of the Third International Conference on Ubiquitous Computing*, (2001), pp. 232-239.

[Mankoff03] Mankoff, J., Dey, A.K., Hsieh, G., Kientz, J., Lederer, S., Ames, M. "Heuristic Evaluation of Ambient Displays," *Proceedings of the Conference on Human Factors and Computing Systems* (2003) pp. 169-176.

[Moore91] Moore, G. *Crossing the Chasm.* New York, NY: Harper Business. 1991.

[Moran01] Moran, T., Dourish, P., "Introduction to This Special Issue on Context –Aware Computing," *Human-Computer Interaction*, vol. 16, 2-4, (2001), pp. 87-97.

[Mullet02]

[Nielsen94] Nielsen, J., Mack R.L., *Usability Inspection Methods*, John Wiley & Sons, Inc., New York (1994).

[Nielsen93] Nielsen, J. *Usability Engineering.* Morgan Kaufmann, San Diego (1993) pp. 135-7, 170.

[Norman88] Norman, D.A., *The Design of Everyday Things*, Currency and Doubleday, New York (1988) pp. 164-5.

[Proctor03] Proctor, R. and Vu, K., "Human Information Processing: An Overview for Human-Computer Interaction," *The Human-Computer Interaction Handbook*, Lawrence Erlbaum Associates, New Jersey (2003), pp. 35-51.

[Reiners99] Reiners, D., Stricker, D., Klinker, G., Müller, S., "Augmented Reality for Construction Tasks: Doorlock Assembly," *Augmented Reality: Placing Artificial Objects in Real Scenes, Proceedings of IWAR'98*, AK Peters, Massachusetts (1999) p.32.

[Scholtz02] Scholtz, J. Human-robot Interactions: Creating Synergistic Cyberforces, in Alan C. Schultz and Lynne E. Parker (eds.), *Multi-Robot Systems: From Swarms to Intelligent Automata*, in Kluwer, 2002.

[Scholtz03] Scholtz, J. and Bahrami, S. Human-Robot Interaction: Development of an Evaluation Methodology for the Bystander Role of Interaction. *Proceedings of the IEEE Conference on System, Man, and Cybernetics, 2003.* Washington, DC. Oct. 2003.

[Shafer01] Shafer, S., Brummitt, B., and Cadiz, JJ. "Interaction Issues in Context-Aware Intelligent Environments," *Human-Computer Interaction*, Vol. 16, 2-4. (2001), pp.363-378.

[Smith86] Smith, S. L. and Mosier, J. N. "Guidelines for designing user interface software," (Technical Report ESD-TR-86-278), Hanscom Air Force Base, MA: USAF Electronic Systems Division (1986)

[Weiser91] Weiser, M., "The Computer for the 21st Century," *Scientific American*, 265, (1991), pp.94-104.